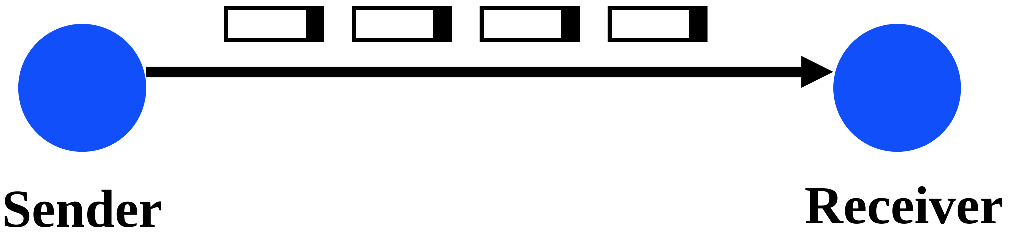
Link Flow Control and Error Control

- Naïve protocol.
- Dealing with receiver overflow: flow control.
- Dealing with packet loss and corruption: error control.
- Meta-comment: these issues are relevant at many layers.
  - » Link layer: sender and receiver attached to the same "wire"
  - » End-to-end: transmission control protocol (TCP) - sender and receiver are the end points of a connection
- How can we implement flow control?
  - » "You may send" (windows, stop-and-wait, etc.)
  - » "Please shut up" (source quench, 802.3x pause frames, etc.)
  - » Where are each of these appropriate?

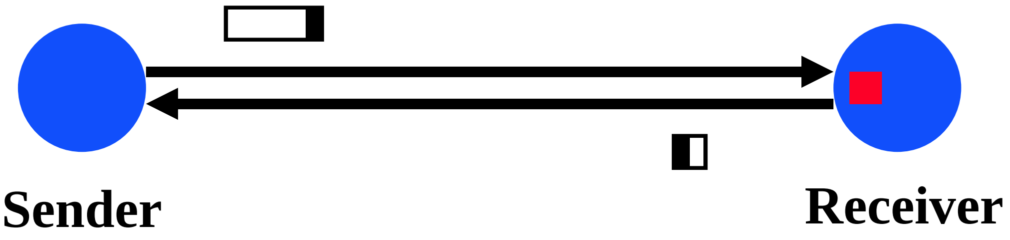
A Naïve Protocol

- Sender simply sends to the receiver whenever it has packets.
- Potential problem: sender can outrun the receiver.
  - » Receiver too slow, buffer overflow, ..
- Not always a problem: receiver might be fast enough.



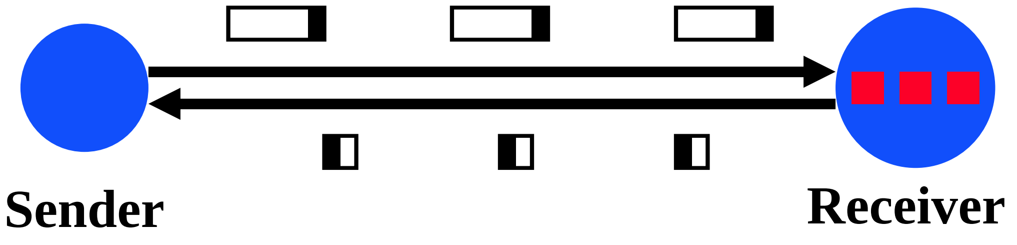
Adding Flow Control

- Stop and wait flow control: sender waits to send the next packet until the previous packet has been acknowledged by the receiver.
  - » Receiver can pace the receiver
- Drawbacks: adds overheads, slowdown for long links.

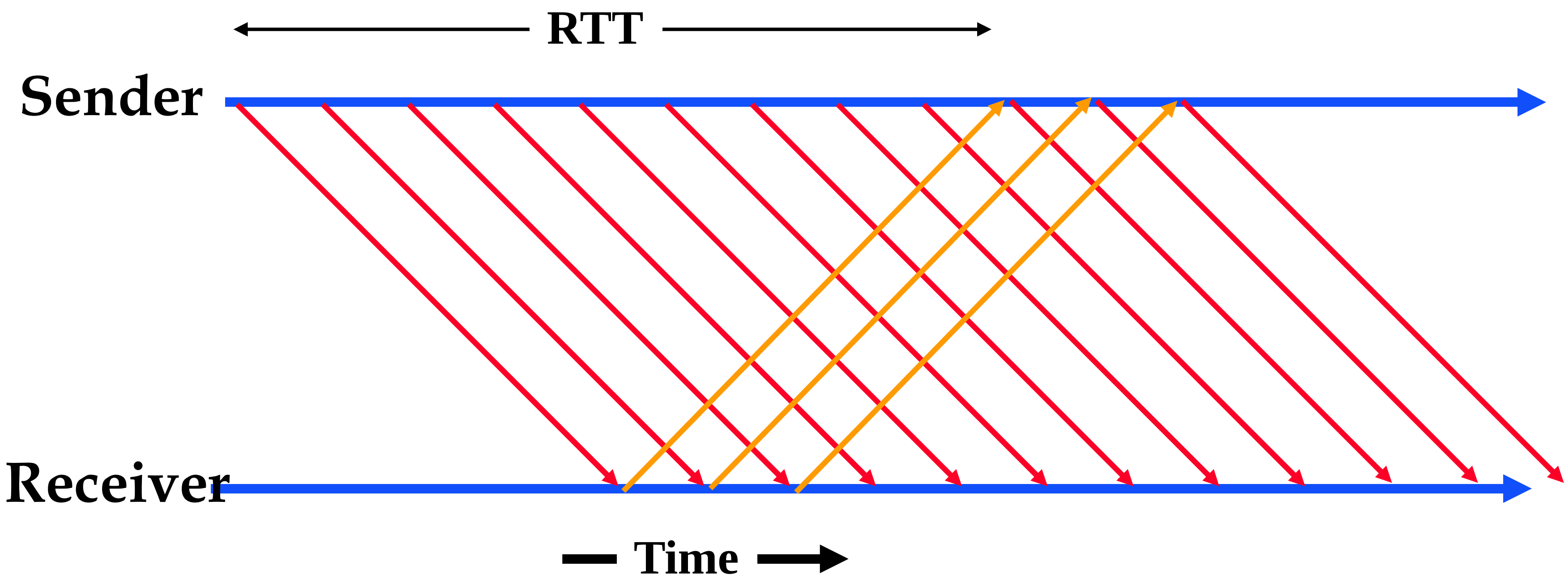


Window Flow Control

- Stop and wait flow control results in poor throughput for long-delay paths: packet size/ roundtrip-time.
- Solution: receiver provides sender with a window that it can fill with packets.
  - » The window is backed up by buffer space on receiver
  - » Receiver acknowledges the a packet every time a packet is consumed and a buffer is freed



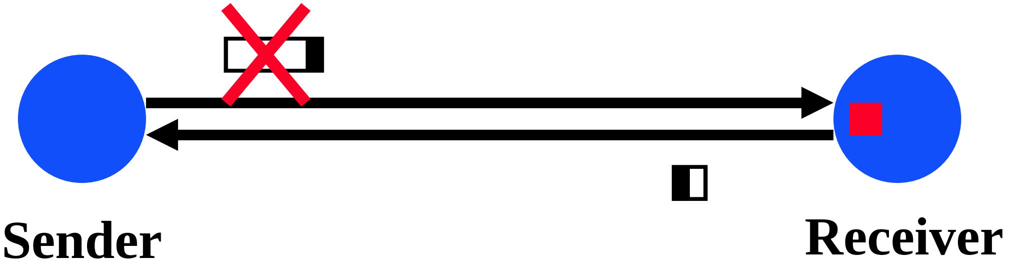
Bandwidth-Delay Product



$$\text{Max Throughput} = \frac{\text{Window Size}}{\text{Roundtrip Time}}$$

Dealing with Errors
Stop and Wait Case

- Packets can get lost, corrupted, or duplicated.
  - » Error detection or correction turns corrupted packet in lost or correct packet
- Duplicate packet: use sequence numbers.
- Lost packet: time outs and acknowledgements.
  - » Positive versus negative acknowledgements
  - » Sender side versus receiver side timeouts
- Window based flow control: more aggressive use of sequence numbers (see transport lectures).



What is Used in Practice?

- No flow or error control.
  - » E.g. regular Ethernet, just uses CRC for error detection
- Flow control only.
  - » E.g. Gigabit Ethernet
- Flow and error control.
  - » E.g. X.25 (older connection-based service at 64 Kbs that guarantees reliable in order delivery of data)